

## تمرین دوم درس پایگاه داده

محمد امانلو - 810100084

استاد شاکری

### سوال 1)

**الف )** کلید کاندیدا کلیدی است که دارای ویژگی های زیر است (یک یا چند تا از Attribute ها که دارای ویژگی های زیر است):

1. اینکه کلید کاندیدا نباید هیچ مقدار تکراری در خود داشته باشد.
  2. باید کمینه باشد. یعنی از مجموعه Attribute های کلید نتوانیم یکی را حذف کنیم و شرط اول همچنان برقرار باشد.
- حال در صورتی که یک مجموعه ای از Attribute ها ویژگی اول از کلید کاندید بودن را داشته باشند اما ویژگی دوم را نداشته باشند. این مجموعه را یک SuperKey یا یک ابرکلید می نامیم.
- از بین کلید های کاندید شده ادمین پایگاه داده یک کلید را مطابق با فضای دامن و دنیای واقعی بعنوان کلید اصلی انتخاب می کند.

**ب) 1.** روش اول این است که این کار را Restrict یا محدود کنیم یعنی تا زمانی که آن View ها وجود دارند ما نمی توانیم که این جداول را حذف کنیم و DBMS از انجام این کار جلوگیری می کند تا وقتی که ما تمام View های مربوطه را پاک کنیم. که این موضوع را در SQL به صورت پرس و جوی زیر نمایش می دهند:

#### **DROP TABLE Employees RESTRICT**

**2.** روش دوم این است که ما بگوییم که پس از حذف جدول تمامی View های مربوط به آن را نیز حذف کن که اصطلاحا این روش را Cascade کردن می نامند. این موضوع را در SQL به صورت پرس و جوی زیر نمایش می دهند:

#### **DROP TABLE Employees CASCADE**

**ج) مشکلات زیادی در حالات مختلف می تواند رخ دهد به همین دلیل نیز اکثر DBMS ها این کار را بسیار محدود کرده اند.**

در صورتی که View ما تنها به یک جدول مرتبط باشد، ما با اعمال تغییرات در جدول اصلی می توانیم تغییرات را بر روی View نیز اعمال کنیم، مثلا در صورتی که می خواهیم یک سطر از view را حذف کنیم می توانیم همان سطر را از جدول پایه ای مربوطه نیز حذف کرد. در صورتی که بخواهیم یک سطر به view اضافه کنیم می بایست آن سطر را به جدول اصلی اضافه کرد. همچنین برای تغییر مقادیر نیز باید مقادیر را در جدول اصلی تغییر داد حال چند مشکل وجود دارد:

1. در صورتی که ما در پی حذف یک سطر از view باشیم و کلید اصلی در view نباشد، آنگاه امکان حذف چندین سطر با مشخصات موجود در view در جدول پایه ای وجود دارد که این موضوع باعث حذف چند داده از جدول می شود.
  2. در صورتی که ما در پی اضافه کردن یک سطر باشیم وقتی به جدول اصلی این سطر را اضافه می کنیم، به دلیل آنکه الزاما تمامی ستون های جدول اصلی در view وجود ندارد ما آن ستون هایی را که در view هستند را با null مقدار دهی می کنیم. حال در صورتی که کلید اصلی جدول در view نباشد، مقدار کلید اصلی باید null شود که غیر ممکن است.
  3. گاهی برخی update ها یا insert ها روی view باعث می شود که یک داده به جدول اصلی اضافه شود اما به view نه مثلا ما یک view از دانشجویان با معدل بیشتر از 17 داریم حال در صورتی که یک دانشجو با معدل 16 بخواهیم به view اضافه کنیم، این دانشجو به جدول پایه ای اضافه می شود اما به view ما اضافه نمی شود. با قرار دادن دستور **WITH CHECK OPTION** می توان از این موضوع جلوگیری کرد.
- اما در صورتی که view ما از چند جدول استفاده کند پیچیده تر نیز می شود:
- به مثال زیر توجه کنید:

Students:

sid	name	major
50000	Dave	CS
53666	Jones	CS
53688	Smith	EE

MajorCourses

major	course
CS	OS
CS	DB

```
CREATE VIEW StudentCourses (sid, major, course)
AS SELECT S.sid, M.major, M.course
FROM Students S, MajorCourses M
WHERE S.major = M.major
```

sid	major	course
50000	CS	DB
50000	CS	OS
53666	CS	DB
53666	CS	OS

حال اگر بخواهیم tuple مثل (CS, DB, 50000) را حذف کنیم یا مقداری از آن را تغییر دهیم هیچ راه حل ساده ای وجود ندارد در واقع در صورتی که ما بخواهیم آن را حذف کنیم. هر مسیری را که طی کنیم این امکان وجود ندارد مثلا اگر ردیف 50000 را از جدول Students حذف کنیم بعنوان اثر جانبی دو سطر از View حذف می شود. در صورتی که مثلا بخواهیم سطر CS DB را از جدول دوم حذف کنیم باز هم دو سطر از View حذف می شود. که برای ما نامطلوب است.

**سوال 2 الف)** خیر این امکان وجود ندارد مگر اینکه ما برای ستون cid یک Value generator داشته باشیم که مقادیری را به طور پیش فرض و به طور unique به آن اختصاص دهد وگرنه چون این سطر دارای Primary key نیست پس ما امکان افزودن یک سطر با کلید اصلی null به جدول را نداریم. برای حل این مشکل باید مقدار cid مربوط به این مشتری را به آن اضافه کنیم.

```
INSERT INTO Customers (cid, cname, nationality, age)
VALUES (1, 'Grant Robert', 32, 'American')
```

(ب)

```
DELETE FROM Customers C WHERE C.age < 18
```

(ج)

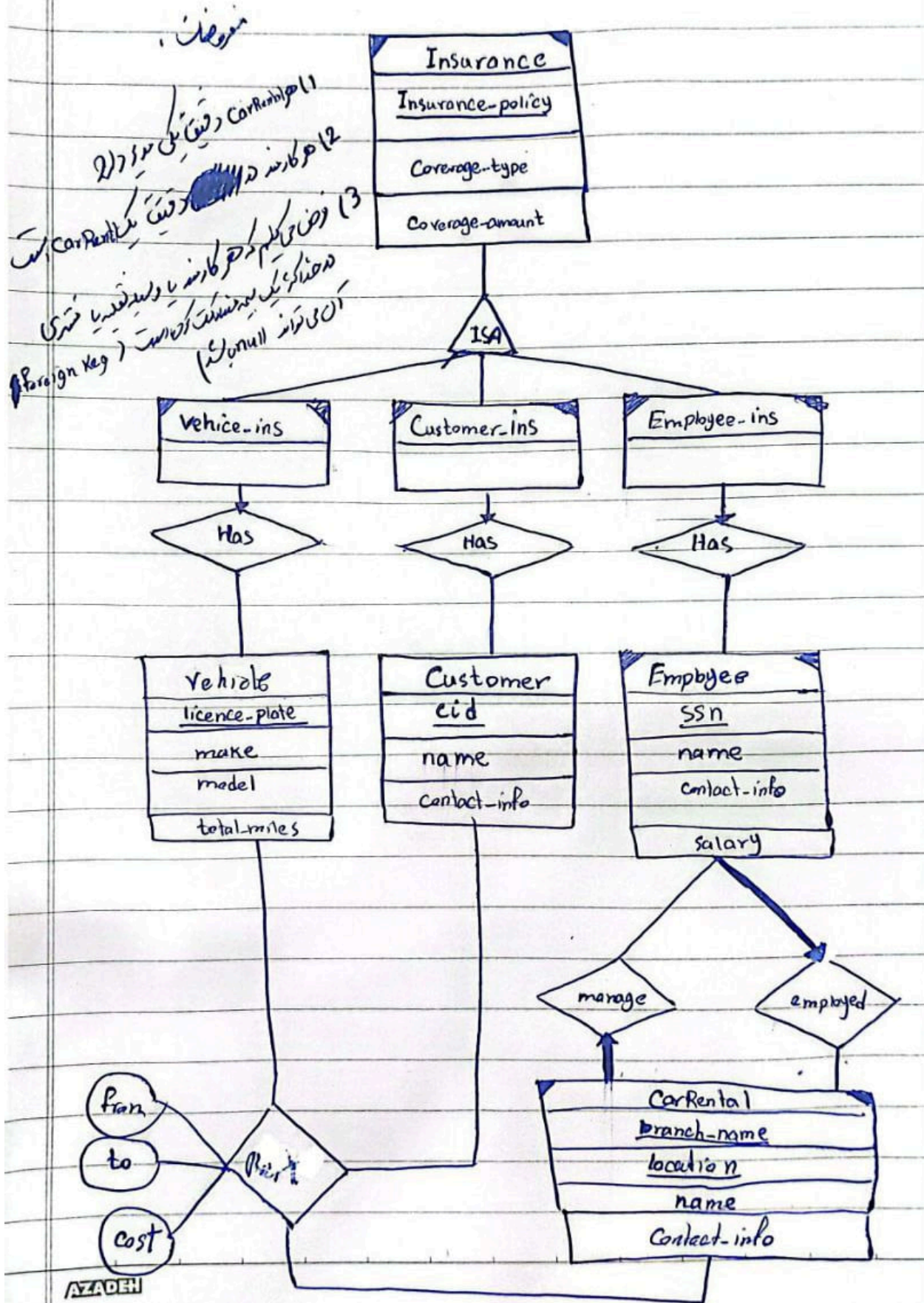
```
UPDATE Hotels H
SET H.rating = 1.2 * H.rating
WHERE H.brand = "Hilton"
```

```
CREATE VIEW MyView(price,brand,cname)
AS SELECT R.price, H.brand, C.name
FROM Reservation R, Hotel H, Customer S
WHERE R.price < 500 AND H.location="Cambridge"

DROP VIEW MyView;
```

ه) در ترکیب اول چون cid و hid بعنوان کلید اصلی در نظر گرفته شده است یعنی هر ترکیب این دو باید یکتا باشند، پس یک مشتری می تواند چند هتل مختلف رزرو کرده باشد (توجه شود که hid همان hotel\_id است). در این حالت اما باید هر مشتری در هر هتل تنها یکبار رفته باشد، یعنی هیچ تکراری در مشتری با همان هتل قابل قبول نیست و تنها یک داده از حضور مشتری در آن هتل را می توان ذخیره کرد. که می تواند مثلا آخرین حضور هر مشتری در هر هتل باشد.

در حالت دوم هر مشتری تنها می تواند یک رزرو در کل جدول داشته باشد در واقع ما تنها آخرین باری که این مشتری در یکی از هتل های ما حاضر بوده را در نظر میگیریم. همچنین به دلیل اینکه hid , تاریخ ورود و خروج به هتل مقدار Unique دارند پس در این حالت امکان اینکه دو مشتری به طور همزمان وارد یک هتل شده باشند و به طور همزمان از هتل خارج شوند نیز وجود ندارد. در واقع هیچ دو رزروی از یک هتل دقیقا برای یک بازه زمانی نباشند.



#### سوال 4)

CREATE TABLE Patient(PID INTEGER, Name CHAR(20), Age INTEGER, Contact\_info CHAR(20),  
PRIMARY KEY PID)

CREATE TABLE Appointment( DID INTEGER, PID INTEGER, Date DATE, Total\_cost REAL,  
PRIMARY KEY (DID,PID), FOREIGN KEY (DID) REFERENCES Doctor\_WorksIn ON DELETE NO  
ACTION, FOREIGN KEY (PID) REFERENCES Patient ON DELETE NO ACTION)

CREATE TABLE Doctor\_WorksIn( DEID INTEGER NOT NULL, DID INTEGER, NAME CHAR(20),  
Contact\_info CHAR(20), License CHAR(20), From Date, To Date, PRIMARY KEY (DID), FOREIGN  
KEY (DEID) REFERENCES Department ON DELETE NO ACTION)

CREATE TABLE Department( DEID INTEGER, Specialization CHAR(20), Manager CHAR(20),  
PRIMARY KEY (DEID) )

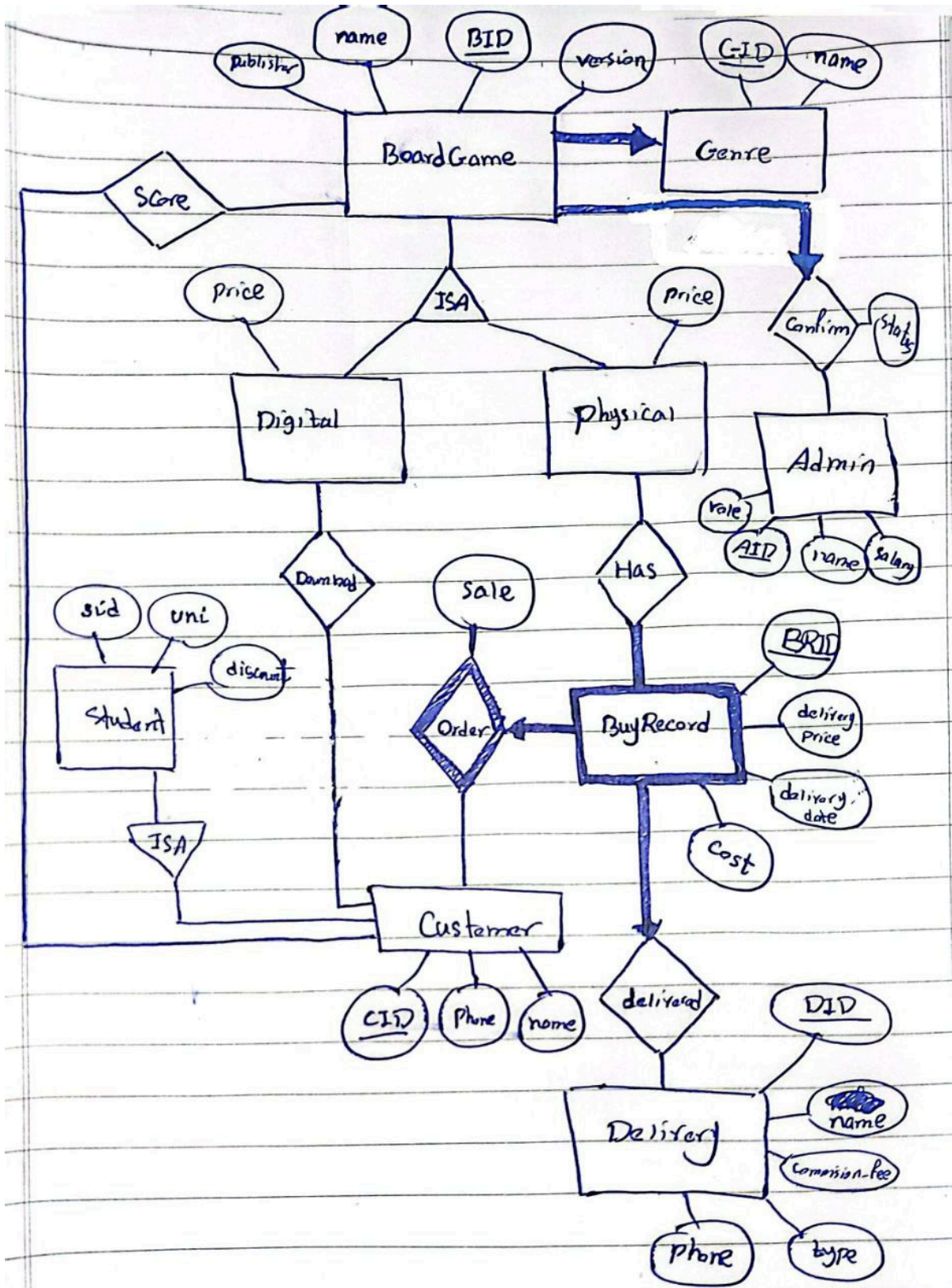
CREATE TABLE Treatment( Dosage INTEGER, Instructions CHAR(20), Medication CHAR(20),  
PRIMARY KEY (Dosage,Instructions,Medication) )

CREATE TABLE MedicalRecord( MID INTEGER, Diagnosis CHAR(20), PID INTEGER NOT NULL,  
DID INTEGER NOT NULL, PRIMARY KEY (MID,PID), FOREIGN KEY (PID) REFERENCES Patient ON  
DELETE CASCADE, FOREIGN KEY (DID) REFERENCES DOCTOR ON DELETE NO ACTION)

در اینجا ما ON DELETE CASCADE گذاشتیم چون weak entity داشتیم.

CREATE TABLE MedicalHAS( MID INTEGER, PID INTEGER, Dosage INTEGER, Instruction  
CHAR(20), Medication CHAR(20), PRIMARY KEY (MID,PID,Dosage,Instruction,Medication),  
FOREIGN KEY (MID,PID) REFERENCES MedicalRecordPatientDoctor ON DELETE NO ACTION,  
FOREIGN KEY (Dosage,Instruction,Medication) REFERENCES Treatment ON DELETE NO  
ACTION )





مفروضات:

هر بوردگیم دقیقا یک ادمین و دقیقا یک ژانر دارد. میشد با توجه به صورت سوال که می گوید هر بورد گیم را یک ادمین تایید کند آن را به صورت حداقل یکی در نظر گرفت اما منطقی تر این است که ما محدودیت دقیقا یکی را اعمال کنیم.

هر رکورد خرید دقیقا یک پستچی دارد.

هر رکورد خرید را دقیقا یک مشتری به ثبت رسانده است و خرید اشتراکی امکان پذیر نیست.

اما هر رکورد خرید حداقل یک بورد گیم فیزیکی را دارد اما می تواند چند بورد گیم باشد.

ما Student را یک Entity جدا فرض میکنیم چون اگر آن را بعنوان Attribute برای مشتریان لحاظ کنیم مصرف حافظه اضافه دارد.